Europäisches Patentamt	European Patent Office	Office européen des brevets

Bescheinigung	Certificate	Attestation
Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.	The attached documents are exact copies of the European patent application described on the following page, as originally filed.	Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr.	Patent application No.	Demande de brevet n°

02090257.3

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

DEN HAAG,DEN
THE HAGUE,   25/07/03
LA HAYE,LE

EPA/EPO/OEB Form   1014   - 02.91

**Europäisches
Patentamt**

**European
Patent Office**

**Office européen
des brevets**

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.:   02090257.3
Demande n°:

Anmeldetag:
Date of filing: 16/07/02
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):

IHP GmbH-Innovations for High Performance Microelectronics/Institut für innovative Mikroelektronik
15236 Frankfurt an der Oder
GERMANY

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:

Synchronizer

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BG/BE/CH/CY/CZ/DE/DK/EE/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

Berlin, 15 July 2002

Our Ref.: IB 1231-01EP   LE/shi
Direct Dial: 030/841 887-0

Applicant: IHP GMBH
Serial Number: New Application

EPO-BERLIN
16 -07- 2002

IHP GMBH – Innovations for High Performance Microelectronics/ Institut für innovative Mikroelektronik
Im Technologiepark 25, D-15236 Frankfurt (Oder)
Synchronizer

BACKGROUND OF THE INVENTION

5   The IEEE 802.11a standard makes use of the Orthogonal Frequency Division Multiplex (OFDM) transmission scheme. The main feature of the OFDM is that the information stream is not transmitted into a single carrier, but is divided into several sub-carriers, each transmitting at a much lower rate. Furthermore, all these sub-carriers are *orthogonal*, i.e. they overlap their spectra but without causing mutual interference.

10   The fact that the different sub-carriers overlap their spectra makes specially difficult one of the main operations at the receiver: synchronization. In this standard, the information is not transmitted continuously, but into bursts. Each burst contains a single frame compound of different OFDM symbols. At the beginning of the frame, four preamble symbols are transmitted.

The synchronization process is data-aided, i.e. is based on the digital processing of the preamble symbols, and is responsible to detect the incoming frame as well as to estimate possible frequency errors and to provide a reference channel estimation to the channel estimation block.

5   We propose a low-power synchronizer structure for the IEEE 802.11a standard, able to estimate frequency offsets in the range ±468 kHz with very simple and effective frame detection and timing synchronization.

OFDM signals are very much sensitive to the synchronizer performance, mainly because the different sub-carriers overlap their respective spectra. The synchronizer

10   is the block responsible for detecting the incoming frame and to estimate and correct for the possible frequency offsets. It also decides the starting point from which on the different OFDM symbols will be fed into the FFT block.

To carry out all of these operations, the standard IEEE 802.11a defines the so-called *preamble symbols*. These symbols have a very specific periodic structure to

15   simplify the estimation procedure and are placed at the very beginning of each frame during transmission.

The following operations have to be carried out:

        Frame detection.
        Carrier frequency offset.
20   Symbol timing estimation.
        Extraction of the reference channel.
        Data reordering.

25

Summary of the Invention

Problems to be solved:

- Synchronization of the OFDM frames, especially in an IEEE 802.11a standard based transmission system.
- Low power realization of the synchronizer.

Novelties:

Optimization of each discrete component of the design.

Optimization of the final architecture by dividing it into different gated clock domains.

During reception, the synchronizer has to peer the channel in order to detect an incoming frame. In our solution, we propose a singular architecture with a novel peak detector algorithm.

The fine timing estimation is realized by a crosscorrelator, with the following features:

This crosscorrelator has been simplified to be 32 samples long (shortest possible length in order to get good results).

Instead of normal complex multipliers, our solution makes use of XNOR-based multipliers.

The 64-point FFT has been divided into a 2-dimensional 8x8-point FFT, allowing an implementation with 60% less complex multipliers compared with the Butterfly solution. This FFT architecture was already patented to us.

Two CORDIC processors are being used for the arctangent calculation as well as the NCO operation. The separation of both operations allows a big simplification in the control mechanism. The main structure of the CORDIC was already patented to us.

Optimization of the final architecture. The proposed architecture is based on the integration of the different components into different clock domains. Each clock domain may be enabled or disabled independently from the others by having a direct control on its clock input signal. This approach reduces the power demands of the whole system.

The present invention is also in each of the following elements or any combination thereof:

A method for detection of the reception of a data frame in an input signal (y_{OFF} (n)), said data frame comprising periodically repeated symbols , comprising the steps of

a)   sampling said input signal (y_{OFF} (n)) with a predetermined sampling rate

b)   transforming said input signal (y_{OFF} (n)) into a first signal ($|J(k)|^2$) that is dependant on an autocorrelation of said input signal with a delayed copy of said input signal, and

c)   detecting a plateau in said first signal ($|J(k)|2$)

d)   generating an output signal that is indicative of detecting said plateau.

A method for detection of the reception of a data frame in an input signal, wherein said transforming step comprises the steps of

-    delaying said input signal by a first predetermined number (N_d) of sampling periods,

-    transforming said input signal into a second signal that is dependant on the complex conjugate of said input signal, and

-    generating a third signal that is dependant on the product of said second signal and of said delayed input signal.

A method for detection of the reception of a data frame in an input signal, comprising a step of saving said third signal for a second predetermined number (N_{avg}) of sampling periods.

A method for detection of the reception of a data frame in an input signal, comprising a step of generating a fourth signal that is dependant on a sum of said second predetermined number (N_{avg}) of third signals.

A method for detection of the reception of a data frame in an input signal, wherein said fourth signal is created by adding said third signal of a current sampling period to said fourth signal of a last previous sampling period and subtracting one third signal, that was saved said second predetermined number (Navg) of sampling periods earlier.

A method for detection of the reception of a data frame in an input signal, wherein said first signal is obtained as the product of said fourth signal and its complex conjugate.

A method for detection of the reception of a data frame in an input signal, wherein said step of detecting a plateau comprises a step of generating a fifth signal that is dependant on the time derivative said first signal.

A method for detection of the reception of a data frame in an input signal, wherein said step of generating said fifth signal comprises a step of delaying said first signal for a third predetermined number of sampling periods, and a step of generating a difference signal that is dependant on the difference between said first signal of a current sampling period and said delayed first signal.

A method for detection of the reception of a data frame in an input signal, comprising a step of detecting an absolute maximum of said fifth signal ($J_{diff}(k)$) within a predetermined range of sampling periods.

A method for detection of the reception of a data frame in an input signal, comprising a step of comparing said fifth signal of said current sampling period with said fifth signal ($J_{diff}(k)$) of a previous sampling period saved in a register, and a step of saving said fifth signal ($J_{diff}(k)$) of said current sampling period to said register, given the condition that its value is larger than that of said fifth signal ($J_{diff}(k)$) of a previous sampling period, thus replacing said earlier fifth signal ($J_{diff}(k)$) in said register under said condition.

A method for detection of the reception of a data frame in an input signal, comprising a step of incrementing a count index by one given the condition that the value of said fifth signal ($J_{diff}(k)$) of said current sampling period is equal or smaller than that of said fifth signal ($J_{diff}(k)$) saved in said register.

A method for detection of the reception of a data frame in an input signal, comprising a step of generating a sixth signal indicative of the condition whether or not the count index has reached a predetermined value.

A method for detection of the reception of a data frame in an input signal, comprising a step of detecting a falling slope in said fifth signal (Jdiff (k)).

A method for detection of the reception of a data frame in an input signal, comprising the steps of

- generating an accumulation signal that is dependant on the sum of said fifth signal (J_{diff} (k)) over a fourth predetermined number of consecutive sampling periods
- comparing said current accumulation signal with the last previous accumulation signal representing without overlap said fourth predetermined number of consecutive earlier sampling periods
- generating a seventh signal indicative of the condition whether or not the value of said current accumulation signal is smaller than the value of said earlier accumulation signal.

A method for detection of the reception of a data frame in an input signal, comprising a step of generating an eighth signal indicative of the condition

- that said sixth signal indicates that said count index has reached said predetermined value and
- that said seventh signal indicates that said value of said current accumulation signal is smaller than said value of said earlier accumulation signal.

A method for detection of the reception of a data frame in an input signal, wherein said output signal is indicative of the time of detecting said plateau.

A method for detection of the reception of a data frame in an input signal, wherein said method is used for detecting a data frame containing OFDM symbols.

A frame detector adapted to performing any of the above methods.

A peak detector for detecting a maximum in a sampled input signal, said peak detector comprising an input port, a peak detection unit communicating with said

input port, and an output port communicating with said peak detection unit, wherein said peak detection unit comprises

a)   a first detection unit connected to said input port and comprising a first memory unit, said first detection unit being adapted to

5   –   comparing said input signal (J_{diff} (k)) received through said input port with a first entry contained in said first memory unit, and to

         –   replacing said first entry by said input signal given the condition that the value of said input signal (J_{diff} (k)) is larger than the value of said first entry,

10   b)   a second detection unit connected to said input port and comprising a second memory unit, said second detection unit being adapted to

         –   generating an accumulation signal, that is dependant on the sum of a current input signal (J_{diff} (k)) and of said fourth predetermined number of previous input signals (J_{diff} (k)),

15   –   comparing said accumulation signal with a second entry contained in said second memory for at least , and to

         –   replacing said second entry by said accumulation signal given the condition that the value of said accumulation signal (J_{diff} (k)) is larger than the value of said second entry,·

20   said peak detection unit being adapted to providing a peak detector output signal at its output port indicative of whether or not said first entry has been unchanged for a predetermined number of sample periods and said second entry has been changed in said current sampling period.

A peak detector for detecting a maximum in a sampled input signal, comprising a
25   counter connected to the output of said first detection unit, said counter being adapted to incrementing a count index given the condition that said value of said accumulation signal (J_{diff} (k)) is equal to or smaller than said value of said second entry.

A peak detector for detecting a maximum in a sampled input signal, wherein said counter is additionally adapted to generating on overflow signal at its output after a fifth predetermined number of consecutive increments.

A peak detector for detecting a maximum in a sampled input signal, wherein said first detection unit comprises a first comparator connected on its input side to said input port and to said first memory unit, and on its output side to a control input of said first memory, said first comparator being adapted to generating a first comparator signal indicative of whether or not said input signal value is larger than said value of said first entry.

A peak detector for detecting a maximum in a sampled input signal, wherein said second detection unit comprises a second comparator receiving on its input side said accumulation signal and said second entry, and on its output side to a control input of said second memory, said second comparator being adapted to generating a second comparator signal indicative of whether or not said accumulation signal value is said accumulation signal value is larger than said value of said second entry.

A peak detector for detecting a maximum in a sampled input signal, comprising an AND-gate receiving at its input side said first comparator signal and a logical inversion of said second comparator signal, and wherein said peak detector output signal is or corresponds to an output signal of said AND-gate.

A method for estimating a relative frequency offset (f_6) in an input signal ($y_{OFF}(n)$), comprising the steps of

a) estimating a coarse frequency offset (β)

b) estimating a fine frequency offset (α) in dependence of said estimated coarse frequency offset (β).

A method for estimating a relative frequency offset (f_6) in an input signal ($y_{OFF}(n)$), wherein said step of estimating said coarse frequency offset (β) comprises a step of transforming said input signal ($y_{OFF}(n)$) into a ninth signal ($|J(k)|^2$) that is dependant on an autocorrelation of said input signal with a delayed copy of said input signal.

A method for estimating a relative frequency offset (f_ε) in an input signal ($y_{OFF}(n)$), wherein said estimating steps of estimating a coarse frequency offeset (β) and/or of calculating a fine frequency offset comprise a step of calculating a phase of said ninth signal ($|J(k)|^2$).

5   A method for estimating a relative frequency offset (f_ε) in an input signal ($y_{OFF}(n)$), wherein said transforming step comprises the steps of

- delaying said input signal by a sixth predetermined number (N_d) of sampling periods,
- transforming said input signal into a tenth signal that is dependant on the

10   complex conjugate of said input signal, and
- generating an eleventh signal that is dependant on the product of said tenth signal and of said delayed input signal.

A method for estimating a relative frequency offset (f_ε) in an input signal ($y_{OFF}(n)$), wherein said sixth predetermined number is chosen such that the ratio between said

15   sixth predetermined number on one side and of the ratio between a sampling frequency and a frequency difference between neighboring subchannels of an Orthogonal Frequency Divisional Multiplexing (OFDM) transmission scheme is an integer value, preferably one.

A method for estimating a relative frequency offset (f_ε) in an input signal ($y_{OFF}(n)$),

20   wherein said phase calculating step comprises a step of calculating an arcus tangens value of a complex conjugate of said ninth signal.

A method for estimating a relative frequency offset (f_ε) in an input signal ($y_{OFF}(n)$), wherein the step of estimating said frequency offset comprises a step of assigning an fine frequency offset value dependant on the value of said coarse frequency

25   offset according to the following function:

$$\varepsilon = \alpha \qquad ; \text{if } (-0.1)/4 \leq \beta \leq (0.1)/4 \qquad (R1)$$

$$\varepsilon = \alpha \qquad ; \text{if } \alpha \geq 0 \text{ and } (0.1)/4 < \beta < (0.9)/4 \qquad (R2)$$

$$\varepsilon = 1 + \alpha \qquad ; \text{if } \alpha < 0 \text{ and } (0.1)/4 < \beta < (0.9)/4 \qquad (R3)$$

$$\varepsilon = 1 + \alpha \qquad ; \text{if } \beta \geq (0.9)/4 \qquad\qquad (R4)$$

$$\varepsilon = -1 + \alpha \qquad ; \text{if } \alpha \geq 0 \text{ and } (-0.9)/4 < \beta < (-0.1)/4 \qquad (R5)$$

$$\varepsilon = \alpha \qquad ; \text{if } \alpha < 0 \text{ and } (-0.9)/4 < \beta < (-0.1)/4 \qquad (R6)$$

$$\varepsilon = -1 + \alpha \qquad ; \text{if } \beta \leq (-0.9)/4 \qquad\qquad (R7)$$

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1:   Shows the preamble symbols used in the IEEE 802.11a standard. It is composed of ten short preamble symbols referred as t_1, ..., t_{10}, each having a length of 0.8 µs, and two long preamble symbols ABCD-ABCD with a double cyclic prefix CD, thus giving the structure CD-ABCD-ABCD.

Figure 2:   This is a scheme of a general delayed autocorrelator. The are two important parameters there: N_d and N_{avg}.

Figure 3:   Detailed scheme of the moving average in the autocorrelator. Each of the delay elements works with a complex sample.

Figure 4:   Results at the output of the delayed autocorrelator for two possible values for N_d keeping $N_{avg} = N_d$. The output $J(k)$ is complex, and so the square magnitude $|J(k)|^2$ is represented. In both cases, several regions showing a plateau are found.

Figure 5:   Structure of the proposed plateau detector, which is separated into a *differentiator* block and a *peak detector* block.

Figure 6:   Sketch of the signal at the output of the differentiator, when the input is as in Figure 4 with $N_d = 64$. Interestingly, a peak is found at the output of this differentiator at that point where the first plateau in

$|J(k)|^2$ starts. This is the main reason why a peak detector algorithm is applied at the output of the differentiator.

Figure 7: Detailed scheme of the several signals involved in the peak detection procedure. The blue signal is given by the group peak detector (falling slope detector). The red signal represents the point where a decision on the existence of a peak is taken.

Figure 8: Particular implementation of the peak detector.

Figure 9: Dependency of the phase of $J^*(k)$ (conjugate value of $J(k)$) with respect to the input carrier frequency offset (normalized) for two particular values of N_d. Afterwards these two values will be used to obtain two estimations for the frequency offset, α and β.

Figure 10: Scheme of the whole frequency offset estimator. There it can be seen that two delayed autocorrelators are being used. Both are used for the frequency offset estimation, but only one of them ($N_d=64$) is used for the frame detection.

Figure 11: In this figure, the different decision regions for the combination of α and β are shown.

Figure 12: Timing of the whole synchronization procedure. This is important to show at which point the input frame has been already corrected for the frequency offset. Only when this correction is done to the input frame, the crosscorrelator can be used for the timing estimation. The figure shows also the portion of the long preamble symbols used as reference signal in the crosscorrelator.

Figure 13: Scheme of the XNOR-based complex multiplier. The two signals to be multiplied are $A=A_{real}+jA_{imag}$ and $B=B_{real}+jB_{imag}$, with $j=\sqrt{-1}$. Both real and imaginary parts are only one bit long, i.e. '1' if the signal is positive and '0' if it is negative.

Figure 14: Possible simplifications of the XNOR-based complex multiplier if the input B is known a priopi. There are four possible cases: $B_{real}=0$,

$B_{imag}=0$; $B_{real}=1$, $B_{imag}=1$; $B_{real}=1$, $B_{imag}=0$; $B_{real}=0$, $B_{imag}=1$. In this case the XNOR gates are simplified to NOT gates, which require a less number of transistors.

Figure 15: Structure of the crosscorrelator used for timing estimation.

Figure 16: Square magnitude of the output of the crosscorrelator when the reference signal is the portion of the long preamble symbols shown in Figure 12. The reference is not directly this portion, but it has been previously complex conjugated and hard-limited.

Figure 17: -General synchronizer scheme. It shows the different clock domains. Each clock domain is activated when it has to carry out some kind of operation. If not, it is disabled in order to reduce the power consumption of the whole synchronizer.

Figure 18: Representation of the two operation modes of the circular CORDIC algorithm being used in the synchronizer. The rotational mode is useful to implement the NCO whereas, the vectoring mode is useful for arctangent calculations.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Frame detection:

During synchronization the reception of a frame is to be detected. This detection is based on the particular periodic structure of the preamble symbols (see Figure 1). The circuit used for this purpose is sketched in Figure 2, where an autocorrelation of the input signal with a delayed version of itself is carried out. Two parameters have been defined there: N_d and N_{avg}. The former (N_d) is the length of the delay and the later (N_{avg}) is the length of the *moving average* block. The moving average consists of the addition of the most recent N_{avg} samples. Its structure is like the one in an FIR filter, but with the advantage that all the coefficients are one. Thus, it is not necessary to add all the N_{avg} samples stored inside the register each time when a new sample comes in, but only add the new sample and subtract the oldest one. The structure is sketched in Figure 3.

The output of the autocorrelation block can be expressed as

$$J(k) = \sum_{l=0}^{N_{avg}-1} y_{OFF}^*(l-k) \cdot y_{OFF}(l-k-N_d) \qquad (1)$$

The selected value for N_d directly depends on the periodicity found in the preamble. From Figure 1, several periodicities are possible, i.e. 16, 32, 48, 64 and 80 samples. Generally, the value for N_{avg} is selected to be the same as N_d.

In Figure 4, the signal $|J(k)|^2$ has been represented for N_d=16 and N_d=64. It can be observed that $|J(k)|^2$ shows a region with a constant value (*plateau*) for a number of samples. The length and position of the plateau directly depends on N_d.

For reasons that we will clarify in the next section, when explaining the carrier frequency offset estimation, we have selected N_d=64 and N_{avg}=N_d. The procedure to detect the frame is based on a plateau detection algorithm for the signal $|J(k)|^2$ (see Figure 4.b).

Making use of a differentiator, the algorithm tries to find out the point where the first plateau starts. At this point, the function $|J(k)|^2$ is not differentiable and an ideal differentiator would show a discontinuity. A real differentiator has a limited bandwidth and so the discontinuity may not happen. In addition, as we are interested in a noise-robust system, this bandwidth has to be as small as possible. As the plateau is 32 samples long, we decided to use the differentiator shown in Figure 5, which is based on a delay line of length 32 together with a subtractor. The output of this subtractor is shown in Figure 6 when the input is the signal of Figure 4.b.

An interesting feature in $J_{diff}(k)$ is that it shows an absolute maximum at the point where the plateau starts (see Figure 6). The autocorrelation block together with the differentiator and the peak detector will constantly peer the channel. When the peak detector identifies an absolute maximum, the synchronizer will consider that a new frame has arrived and the carrier frequency offset estimator will be activated. Because of the noise (thermal, digital), the peak detection will not be a trivial task, i.e. a smart peak detection algorithm will be necessary in order to distinguish the absolute from the relative maxima.

The peak detection block is divided into two sub-blocks: *group peak detector* and *instantaneous peak detector.*

The instantaneous peak detector is composed of a comparator and a counter. The present sample $J_{diff}(k)$ coming out from the differentiator is compared with the last recorded maximum J_{max}. As long as the sample $J_{diff}(k)$ is bigger than J_{max}, the register storing J_{max} will be updated to contain the new sample $J_{diff}(k)$ as the latest maximum and the counter will be reset.

If $J_{diff}(k)$ is smaller or equal than J_{max}, the counter will be triggered and it will increase its count by one. If this situation remains until the counter counts through its full range, the instantaneous peak detector will generate a signal stating that a relative peak was found inside the counting scope of the counter. In this implementation a 4-bit counter was used, which makes a counting scope of 16.

The group peak detector is used to detect falling slopes in $J_{diff}(k)$, and its main component is also a comparison block. If the group peak detector finds a falling edge at the same time as the instantaneous peak detector finds a relative peak, this means that the detected peak is actually an absolute peak.

In the group peak detector, the input signal is accumulated in groups of six samples (6-tuples) and the present group is compared with the previous one. If it is smaller, it means that the falling slope has started.

The results of the peak detection algorithm have been sketched in Figure 7, whilst a general block diagram of the peak detector is shown in Figure 8.

We have to mention that the frame will be detected 16 samples after its actual starting point, i.e. the detection will occur in the middle of the first plateau in $|J(k)|^2$. However, this fact does not pose any problem, and the reason for the same will be given in the next section, where we explain the carrier frequency offset estimation.

Carrier frequency offset estimation and correction:

During the RF down-conversion, the local oscillator (LO) normally is not exactly tuned to the expected frequency, but it will show some offset. As the different sub-carriers in an OFDM system overlap, any frequency offset in the receiver side may

lead to significant Inter-Carrier-Interference (ICI). Once the synchronizer has been fired after frame detection, the next operation will be the estimation of this frequency offset. Here we recover the expression given in (1) for the autocorrelation $J(k)$, considering that

$$y_{OFF}(n) = y(n) \cdot e^{j2\pi f_\varepsilon \frac{\Delta f}{f_s} n} \tag{2}$$

where $y_{OFF}(n)$ is the signal affected by a normalized frequency offset f_ε. This normalization is with respect to the channel spacing Δf in the OFDM signal, which is 312.5 kHz in the IEEE 802.11a standard. The parameter f_S is the inverse of the FFT time ($f_S = 20$ MHz in the standard under consideration). The signal $J(k)$ may then be rewritten as

$$J(k) = e^{-j2\pi f_\varepsilon \frac{\Delta f}{f_s} N_d} \cdot \sum_{l=0}^{N_{avg}-1} y^*(l-k) \cdot y(l-k-N_d) \tag{3}$$

If $y(n)$ is a periodic signal with a period of N_d samples, i.e. $y(n) = y(n-N_d)$, then (3) can be simplified to

$$J(k) = e^{-j2\pi f_\varepsilon \frac{\Delta f}{f_s} N_d} \cdot \sum_{l=0}^{N_{avg}-1} |y(l-k)|^2 \tag{4}$$

From equation (4) it can be seen that the phase of $J(k)$ is only due to f_ε, and so f_ε could be found as follows

$$f_\varepsilon = \frac{f_S}{2\pi \cdot N_d \cdot \Delta f} \tan^{-1}\left(J^*(k)\right) \tag{5}$$

However, there are several factors which destroy the periodicity, making $y(n) \neq y(n-N_d)$. The most important ones are the AGC settling time and the channel impulse response. Another factor is noise, but its effect can be largely compensated by the averaging N_{avg} in (4). In addition, if N_{avg} is a multiple of the minimum periodicity in the preambles (16 samples in this standard), $|J(k)|^2$ shows a plateau in the region where the phase of $J(k)$ only depends on the carrier frequency offset.

Considering what has been stated above, the frequency offset could be estimated after calculation of the phase of $J(k)$, at that point k where the peak detector established the beginning of the frame, because this point falls exactly in the middle of the plateau in $|J(k)|^2$.

Nevertheless, there is a limitation in the frequency offset estimation, which can be obtained by calculating the phase in expression (4), i.e.

$$-\pi \leq 2\pi f_\varepsilon \frac{\Delta f}{f_s} N_d < \pi \tag{6}$$

The ratio $\Delta f / f_s$ is 1/64 in the IEEE 802.11a standard. From (6) it can be seen that the range of possible estimated values for f_ε will only depend on the selected delay N_d in the autocorrelator. The dependency of the frequency estimation with respect to the actual frequency offset is shown in Figure 9 for two particular values of N_d.

In our implementation, f_ε is supposed to have its value in the range ±1.5, i.e. a frequency offset of ±468.75 kHz in an oscillator working in the 5 GHz ISM band. If the oscillator is tuned at 5.6 GHz, this value of expected offset is about 85 ppm (parts-per-million).

We will estimate f_ε decomposing its value into two components α and β, and the actual estimation ε will be a function of these two components, i.e. $\varepsilon = f(\alpha, \beta)$. The value of α is restricted to be in the range ±0.5, and so is called the *fine frequency offset*. The parameter β is called *coarse frequency offset*, and it will be in the range ±2.0 (see Figure 9). Our aim is to get proper estimations for these two parameters and combine them to finally obtain ε.

With the presented scheme used for frame detection it is possible to derive the value for α. In this scheme an autocorrelator with N_d=64 is used. We may estimate the value for α by calculating the phase of the output of this autocorrelator, $J_F(k)$. The estimation of β can be obtained using another autocorrelator with N_d=16 and $N_{avg}=N_d$ and calculating also the phase of its output $J_C(k)$ (see Figure 10).

The dependency of α and β vs. f_ε was sketched in Figure 9. There it can be see that this dependency is not completely lineal, but shows some discontinuities. The

reason for these discontinuities are the phase leaps from $+\pi$ to $-\pi$ at the complex exponential in (4). Thus, the final value for ε cannot be a lineal combination of the two estimations α and β. Thus one has to find a suitable way to combine α and β in order to determine ε.

5   From Figure 9 it can also be seen that β has no discontinuities throughout the entire range of expected values for f_ε (± 1.5). Nevertheless, β will be much more noisy than α since the moving average considers only 16 samples (see Figure 10). For this reason, β will provide a coarse approximation of the frequency offset, which will be further refined using α.

10   The selected function $\varepsilon = f(\alpha, \beta)$ is as follows

$$\varepsilon = \alpha \qquad ; \text{if } (-0.1)/4 \leq \beta \leq (0.1)/4 \qquad \text{(R1)}$$

$$\varepsilon = \alpha \qquad ; \text{if } \alpha \geq 0 \text{ and } (0.1)/4 < \beta < (0.9)/4 \qquad \text{(R2)}$$

$$\varepsilon = 1 + \alpha \qquad ; \text{if } \alpha < 0 \text{ and } (0.1)/4 < \beta < (0.9)/4 \qquad \text{(R3)}$$

$$\varepsilon = 1 + \alpha \qquad ; \text{if } \beta \geq (0.9)/4 \qquad \text{(R4)}$$

15   $$\varepsilon = -1 + \alpha \qquad ; \text{if } \alpha \geq 0 \text{ and } (-0.9)/4 < \beta < (-0.1)/4 \qquad \text{(R5)}$$

$$\varepsilon = \alpha \qquad ; \text{if } \alpha < 0 \text{ and } (-0.9)/4 < \beta < (-0.1)/4 \qquad \text{(R6)}$$

$$\varepsilon = -1 + \alpha \qquad ; \text{if } \beta \leq (-0.9)/4 \qquad \text{(R7)} \qquad \text{(7)}$$

In this expression, certain regions have been defined for different values of α and β (see Figure 11). The estimation ε of the frequency offset will be assigned depending 20   on these particular regions.

The blocks used to obtain α and β perform an *arctangent* calculation. This complex mathematical operation can be efficiently realized by using the CORDIC algorithm working in the *vectoring mode*. Although two arctangent blocks have been shown in Figure 10, only one is necessary in the actual implementation. After the frame has 25   been detected (indicated by the peak detector in Figure 10), the two samples $J_C(k)$ and $J_F(k)$ will be stored in a register. The arctangent block will first calculate α from

$J_F(k)$ and afterwards β from $J_C(k)$. More details on the CORDIC algorithm and its implementation can be found in the German Patent 101 64 462.0.

The correction of the frequency offset is carried out considering the signal model given in (2). To obtain the original signal y(n), the input signal $y_{OFF}(n)$ has to be multiplied by a phasor, which is the complex conjugate of the one found in (2). This operation will be carried out by a Numerically Controlled Oscillator (NCO), which is implemented using again the CORDIC algorithm, this time operating in the *rotational mode*.

Symbol timing estimation:

Unlike to what was done during carrier frequency offset estimation, where the periodicity of the *short preamble symbols* was the main feature used for the estimation, the symbol timing estimation will be obtained by exploiting the direct knowledge of the *long preamble symbols*. After the frame is detected, the synchronizer knows approximately which samples of the preamble have entered to the delay line of the autocorrelator, but this knowledge is not enough for further processing and has to be refined.

The main block of the symbol timing estimator is a crosscorrelator. Its main purpose is to compare the input frame with a reference signal, which is directly obtained from the long preamble symbol. The crosscorrelation can only be applied once the samples of the input frame have been corrected for the frequency offset by the NCO.

The "piece" of the long preamble symbols selected as the crosscorrelator reference $c_{REF}(n)$ is shown in Figure 12. The reference has a length of 32 complex samples, which is the shortest possible length for this reference in order to obtain appropriate results after crosscorrelation.

Considering the standard IEEE 802.11a, the reference is as follows: $c_{REF}(0..31)$ = {0.1563, $-0.0051-j0.1203$, $0.0397-j0.1112$, $0.0968+j0.0828$, $0.0211+j0.0279$, $0.0598-j0.0877$, $-0.1151-j0.0552$, $-0.0383-j0.1062$, $0.0975-j0.0259$, $0.0533+j0.0041$, $0.0010-j0.1150$, $-0.1368-j0.0474$, $0.0245-j0.0585$, $0.0587-j0.0149$, $-0.0225+j0.1607$, $0.1192-j0.0041$, $0.0625-j0.0625$, $0.0369+j0.0983$, $-0.0572+j0.0393$, $-0.1313+j0.0652$, $0.0822+j0.0924$,

0.0696+j0.0141,   –0.0603+j0.0813,   –0.0565–j0.0218,   –0.0350–j0.1509,
–0.1219–j0.0166,   –0.1273–j0.0205,   0.0751–j0.0740, .   –0.0028+j0.0538,
–0.0919+j0.1151, 0.0917+j0.1059, 0.0123+j0.0976}, where $j=\sqrt{-1}$.

From the implementation point of view, the complex crosscorrelator is usually a "weak" point in modern communication circuit designs because of its computation complexity (they require a large number of complex multipliers) and subsequent need for large silicon area. Having this in mind, in this implementation we used a simplified scheme for the crosscorrelator, based on simple XNOR 1-bit multipliers (Figure 13), that substitute the commonly used complex multipliers. Instead of multiplying N-bit complex numbers, the XNOR multiplier performs only the multiplication of the sign bits of the complex input values.

In addition, a further simplification in the structure of these multipliers is possible if one of the inputs is fixed and is known beforehand. In this simplification, the XNOR gates may be replaced by NOT gates, which require a less number of transistors (see Figure 14).

The final structure of the crosscorrelator is shown in Figure 15. The structure of each multiplier is decided upon the reference signal $c_{REF}(31..0)^*$, where we already considered the reference to be complex conjugated, hard-limited and order-reversed. The final reference looks as follows

$c_{REF}(31..0)^*$={1, 1, 0, 0, 1+j, j, j, j, j, 0, 1, 1, 0, 0, 1, 1+j, 1+j, 0, 1+j, 1+j, j, 1+j, 1, 1+j, j, j, 1+j, 1, 1, 1+j, j, 1}, where $j=\sqrt{-1}$.

The output of the crosscorrelator is shown in Figure 16, where two major peaks become visible at instants n_{init1} and n_{init2}. Both peaks will happen when the portions AB in the long preamble symbols (see Figure 12) are inside the crosscorrelator. For our purpose it is enough to detect the first peak by setting a certain threshold at the output of the crosscorrelator. The 64 samples coming immediately after this first peak will be fed into the FFT (CDAB fields marked with an orange background in Figure 12) in order to get the *reference channel estimation*.

Reference channel extraction:

The reference channel estimation is used by the channel estimator in order to correct for the filtering due to the transmission channel. The reference channel will be obtained by calculating the FFT of the CDAB fields in Figure 12. This operation may start immediately after the crosscorrelator decides for the initial timing n_{init1}.

5 Nevertheless, the resulting FFT calculation has to be multiplied by the sequence ($-1)^k$, k being the frequency variable. In the IEEE 802.11a standard, the actual long preamble symbols are defined as ABCD, i.e. a cyclic delay of 32 samples into a sequence of 64 samples is introduced. After performing the FFT, any time delay is seen as a linear phase, which in this specific case is reduced to the sequence exp{–

10 $j2\pi(32/64)k\}=\exp\{-j\pi k\}=(-1)^k$. In order to compensate for this linear phase, the FFT output has to be multiplied exactly by this sequence.

The phase compensation mentioned above is only necessary when calculating the reference channel estimation. For the data symbols coming after the preamble no phase correction will be necessary.

15 There are two further operations to be performed inside the FFT block in Figure 17. As the synchronizer already knows the timing of the input samples, it can detect the end of the preamble symbols and the beginning of the data symbols. The data symbols will go through a *cyclic prefix* extraction block prior to the FFT calculation. The cyclic prefix extraction is mainly formed by a counter. For each data symbol, 80

20 samples are expected. The first 16 will be discarded and the last 64 will be fed into the FFT. The cyclic prefix is inserted in the OFDM symbols in order to prevent the Inter-Symbol-Interference (ISI) caused by the channel filtering.

The last operation included inside the FFT block is the channel reordering. Immediately after the FFT calculation, the output samples are delivered in the serial

25 form according to the natural order. For further processing, this order has to be changed. Specifically, the data after FFT is given in the following order: D_0, D_1, D_2, ..., D_{31}, D_{-32}, D_{-31}, D_{-30}, ..., D_{-1}, where the subindex indicates the corresponding subchannel. But the samples D_0, D_{27}, D_{28}, D_{29}, D_{30}, D_{31}, D_{-32}, D_{-31}, D_{-30}, D_{-29}, D_{-28} and D_{-27} (11 samples in total) carry no information and are directly discarded. The

30 remaining 52 samples $\{D_k \ / \ k \in [1,26] \cup [-26,-1]\}$ are provided at the output of the FFT block in the following order: D_{-21}, D_{-7}, D_7, D_{21}, D_{-26}, D_{-25}, ..., D_{-22}, D_{-20}, D_{-19}, D_{-18}, ..., D_{-8}, D_{-6}, D_{-5}, ..., D_{-1}, D_1, ..., D_5, D_6, D_8, ..., D_{18}, D_{19}, D_{20}, D_{22}, ..., D_{25}, D_{26}.

The FFT processor:

The actual implementation of the FFT processor itself is not going to be explained in detail here. The main idea behind this implementation has been a mathematical manipulation of the definition of the FFT in order to convert the 64-point FFT into a 2-dimensional 8x8-point FFT. For more details see the German Patent 100 62 759.5.

The CORDIC processor:

The CORDIC processor is a part of the synchronizer. The processor performs the circular CORDIC algorithm in both of its operation modes, viz. *rotational* and *vectoring*. While the rotation mode of operation of the CORDIC enables us to compute the multiplication of any quantity with a phasor $\exp\{j\varphi\}=\cos(\varphi)+j\sin(\varphi)$, the vectoring mode can be used for computing the magnitude and the phase angle of a complex value. The operation principle of these two modes is shown in Figure 18.

However, for the realization of the synchronizer, these two modes of operation of the CORDIC are independently utilized in two different phases. At first, the phases of $J_F(k)$ and $J_C(k)$ (see Figure 10) are computed (vectoring mode of operation) to obtain α and β, respectively. After calculation of ε by combining α and β, this phase angle is used to obtain a phase correction for the incoming data symbols applying an NCO (rotation mode of operation). The phase angle evaluation takes place only once at the beginning of the frame (after frame detection) whereas, for the rest of the frame the NCO operation is carried out. Thus, in our consideration, it is more pragmatic to separate the two modes of CORDIC operation and realize them in the form of two separate modules. This separation of operations opens up the possibility for applying clock gating to save the power corresponding to the NCO while the computation of the phase angle is carried on and vice versa. On the other hand, this separation results in a massive reduction of control hardware as no reuse of the component takes place that subsequently eliminates any feedback path. Another advantage of separating these two functionalities is that, in this case, it is possible to implement each of them in a much simplified and efficient manner.

When using the rotational CORDIC as an NCO, the input signal is multiplied by a phasor with the form $\exp\{j(2\pi/64)\cdot\varepsilon\cdot n\}$, where ε is the estimated normalized carrier

frequency offset (see Figure 10) and n is the time variable. Thus, the variable φ in Figure 18.a depends on n, and it is updated at each clock cycle. This is done by adding a phase accumulator to the input φ in the CORDIC processor.

We have to mention here that the value of ε must not be sign-reversed in order to apply a correct phase correction, because the result coming from the arctangent calculation already considers this sign. This can be easily seen in expression (4), where by definition, the phase contribution in $J(k)$ already contains the $-ve$ sign.

The actual implementation of the rotational and vectoring CORDIC processors is described in more detail in the German Patent 101 64 462.0.

Advantages of the proposed synchronizer:

The proposed synchronizer has been designed as a power efficient system. This has been achieved in one side by optimizing each block independently and on the other side by dividing the whole synchronizer structure into different clock domains (see Figure 17).

A clock demain separation helps to save power in the sense that only certain regions of the system are activated for operation, while others are not, by applying the *clock* gating. Thus, as shown in Figure 17, three clock domains are used here. The blocks belonging to the clock domain #1 peer the channel trying to detect an incoming frame through the peak detector algorithm. When this is true, the arctangent calculation is triggered, but it will operate only on two samples, thus obtaining two single values for α and β. Afterwards this clock domain is disabled. The combination of α and β to finally obtain ε can be done using combinatorial logic, thus requiring no clock at all.

Disabling the clock domain #1, it is possible to save the power that otherwise would be consumed by the huge delay line at the input of the autocorrelator as well as the FIR structures of the moving average blocks.

Once the value for ε is available, the NCO will start its operation until the end of the frame. The crosscorrelator will be activated at the same time, but it will operate only until a peak is found at its output, being afterwards disabled. This is achieved by assigning a particular clock domain to the crosscorrelator (clock domain #3).

The FFT block is activated after a peak is detected at the output of the crosscorrelator and will operate like the NCO, until the end of the incoming frame, for that reason both the NCO and the FFT blocks belong to the same clock domain #2.

5 Apart from the power saving in the system level by applying clock gating, even more power saving is achieved by optimising each single block in the synchronizer.

Thus, the frame detection algorithm is based on the knowledge of the ideal shape of the signal $|J_F(k)|^2$. A simple and robust peak detection algorithm is applied to this signal in order to detect an incoming frame. The arctangent calculation as well as

10 the NCO are based on the CORDIC algorithm. The CORDIC processors designed for these purposes have been optimised to reach the final angle in an adaptive way and thereby executing a minimum number of iteration steps. Furthermore, the crosscorrelator has been simplified to use XNOR-based complex multipliers instead of the normal complex multipliers and the reference signal therein has been shorten

15 as much as possible in order to obtain valid results. Last but not least, the FFT processor has been also optimised by using a new architecture, which requires no coefficient storing or complex multiplier.

The whole structure introduces a latency of 3.9 µs counted from the instant when the frame is detected. This value is less than one OFDM symbol period (4 µs),

20 which means that no extra storage of the input samples has to be done inside the synchronizer.

In summary, the invention comprises:

The algorithm used for the frame detection, making use of a simplified differentiator to obtain an absolute maximum in the differentiated signal at that

25 point where the first plateau in $J_F(k)$ starts (output of the autocorrelator with $N_d=64$);

The design of the peak detector to obtain the position of the absolute maximum in the differentiated signal, dividing the problem into relative peak detection and falling edge detection;

30 The way to combine the two frequency offsets α and β to finally obtain ε;

The use of a 32-sample long reference signal in the crosscorrelator for timing estimation;

The use of our simplified XNOR-based crosscorrelator, and the simplifications therein based on the knowledge of the reference;

The use of our particular solution for the CORDIC algorithm in the vectoring mode for arctangent calculation;

The use of our particular solution for the CORDIC algorithm as NCO for the frequency offset correction;

The hardware structuring of the whole synchronizer, allowing a very simple control mechanism and the separation of this structure into different clock domains, each one being activated only to perform its operation and deactivated afterwards.

Claims

1. A method for detection of the reception of a data frame in an input signal (y_{OFF} (n)), said data frame comprising periodically repeated symbols ,

   comprising the steps of

   a) sampling said input signal (y_{OFF} (n)) with a predetermined sampling rate

   b) transforming said input signal (yOFF (n)) into a first signal ($|J(k)|2$) that is dependant on an autocorrelation of said input signal with a delayed copy of said input signal, and

   c) detecting a plateau in said first signal ($|J(k)|2$)

   d) generating an output signal that is indicative of detecting said plateau.

2. A method according to claim1, wherein said transforming step comprises the steps of

   - delaying said input signal by a first predetermined number (N_d) of sampling periods,

   - transforming said input signal into a second signal that is dependant on the complex conjugate of said input signal, and

   - generating a third signal that is dependant on the product of said second signal and of said delayed input signal.

3. A method according to claim 2, comprising a step of saving said third signal for a second predetermined number (N_{avg}) of sampling periods.

4. A method according to claim 2 or 3, comprising a step of generating a fourth signal that is dependant on a sum of said second predetermined number (N_{avg}) of third signals.

5. A method according to claim 4, wherein said fourth signal is created by adding said third signal of a current sampling period to said fourth signal of a last previous sampling period and subtracting one third signal, that was saved said second predetermined number (N_{avg}) of sampling periods earlier.

6. A method according to claim 5, wherein said first signal is obtained as the product of said fourth signal and its complex conjugate.

7. A method according to any one of the preceding claims, wherein said step of detecting a plateau comprises a step of generating a fifth signal that is dependant on the time derivative said first signal.

8. A method according to claim 7, wherein said step of generating said fifth signal comprises a step of delaying said first signal for a third predetermined number of sampling periods, and a step of generating a difference signal that is dependant on the difference between said first signal of a current sampling period and said delayed first signal.

9. A method according to claim 7 or 8, comprising a step of detecting an absolute maximum of said fifth signal ($J_{diff}(k)$) within a predetermined range of sampling periods.

10. A method according to claim 9, comprising a step of comparing said fifth signal of said current sampling period with said fifth signal ($J_{diff}(k)$) of a previous sampling period saved in a register, and a step of saving said fifth signal ($J_{diff}(k)$) of said current sampling period to said register, given the condition that its value is larger than that of said fifth signal ($J_{diff}(k)$) of a previous sampling period, thus replacing said earlier fifth signal ($J_{diff}(k)$) in said register under said condition.

11. A method according to claim 10, comprising a step of incrementing a count index by one given the condition that the value of said fifth signal ($J_{diff}(k)$) of said current sampling period is equal or smaller than that of said fifth signal ($J_{diff}(k)$) saved in said register.

12. A method according to claim 11, comprising a step of generating a sixth signal indicative of the condition whether or not the count index has reached a predetermined value.

13. A method according to any one of claims 9 to 12, comprising a step of detecting a falling slope in said fifth signal ($J_{diff}(k)$).

14. A method according to claim 13, comprising the steps of

- generating an accumulation signal that is dependant on the sum of said fifth signal (J_{diff} (k)) over a fourth predetermined number of consecutive sampling periods

- comparing said current accumulation signal with the last previous accumulation signal representing without overlap said fourth predetermined number of consecutive earlier sampling periods

- generating a seventh signal indicative of the condition whether or not the value of said current accumulation signal is smaller than the value of said earlier accumulation signal.

15. A method according to claims 13 and 14, comprising a step of generating an eighth signal indicative of the condition

- that said sixth signal indicates that said count index has reached said predetermined value and

- that said seventh signal indicates that said value of said current accumulation signal is smaller than said value of said earlier accumulation signal.

16. A method according to any one of the preceding claims, wherein said output signal is indicative of the time of detecting said plateau.

17. A method according to any one of the preceding claims, wherein said method is used for detecting a data frame containing OFDM symbols.

18. A peak detector for detecting a maximum in a periodically sampled input signal, said peak detector comprising an input port, a peak detection unit communicating with said input port, and an output port communicating with said peak detection unit, wherein said peak detection unit comprises

a) a first detection unit connected to said input port and comprising a first memory unit, said first detection unit being adapted to

- comparing said input signal (J_{diff} (k)) received through said input port with a first entry contained in said first memory unit, and to

- replacing said first entry by said input signal given the condition that the value of said input signal (J_{diff} (k)) is larger than the value of said first entry,

b) a second detection unit connected to said input port and comprising a second memory unit, said second detection unit being adapted to

- generating an accumulation signal, that is dependant on the sum of a current input signal (J_{diff} (k)) and of said fourth predetermined number of previous input signals (J_{diff} (k)),
- comparing said accumulation signal with a second entry contained in said second memory for at least , and to
- replacing said second entry by said accumulation signal given the condition that the value of said accumulation signal (J_{diff} (k)) is larger than the value of said second entry,

said peak detection unit being adapted to providing a peak detector output signal at its output port indicative of whether or not said first entry has been unchanged for a predetermined number of sample periods and said second entry has been changed in said current sampling period.

19. A peak detector according to claim 18, comprising a counter connected to the output of said first detection unit, said counter being adapted to incrementing a count index given the condition that said value of said accumulation signal (J_{diff} (k)) is equal to or smaller than said value of said second entry.

20. A peak detector according to claim19, wherein said counter is additionally adapted to generating on overflow signal at its output after a fifth predetermined number of consecutive increments.

21. A peak detector according to any one of the claims 18 to 20, wherein said first detection unit comprises a first comparator connected on its input side to said input port and to said first memory unit, and on its output side to a control input of said first memory, said first comparator being adapted to generating a first comparator signal indicative of whether or not said input signal value is larger than said value of said first entry.

22. A peak detector according to any one of the claims 18 to 21, wherein said second detection unit comprises a second comparator receiving on its input side said accumulation signal and said second entry, and on its output side to a control input of said second memory, said second comparator being adapted to generating a second comparator signal indicative of whether or not said accumulation signal value is said accumulation signal value is larger than said value of said second entry.

23. A peak detector according to claims 21 and 22, comprising an AND-gate receiving at its input side said first comparator signal and a logical inversion of said second comparator signal, and wherein said peak detector output signal is or corresponds to an output signal of said AND-gate.

24. A method for estimating a relative frequency offset (f_e) in an input signal ($y_{OFF}(n)$) comprising the steps of

    a)   estimating a coarse frequency offset (β)

    b)   estimating a fine frequency offset (α) in dependence of said estimated coarse frequency offset (β).

25. A method according to claim24, wherein said step of estimating said coarse frequency offset (β) comprises a step of transforming said input signal ($y_{OFF}(n)$) into a ninth signal ($|J(k)|^2$) that is dependant on an autocorrelation of said input signal with a delayed copy of said input signal.

26. A method according to claim 25, wherein said estimating steps of estimating a coarse frequency offeset (β) and/or of calculating a fine frequency offset comprise a step of calculating a phase of said ninth signal ($|J(k)|^2$).

27. A method according to claim 25 or 26, wherein said transforming step comprises the steps of

    -     delaying said input signal by a sixth predetermined number (N_d) of sampling periods,

    -     transforming said input signal into a tenth signal that is dependant on the complex conjugate of said input signal, and

- generating an eleventh signal that is dependant on the product of said tenth signal and of said delayed input signal.

28. A method according to claim 27, wherein said sixth predetermined number is chosen such that the ratio between said sixth predetermined number on one side and of the ratio between a sampling frequency and a frequency difference between neighboring subchannels of an Orthogonal Frequency Divisional Multiplexing (OFDM) transmission scheme is an integer value, preferably one.

29. A method according to anyone of the claims 24 to 28, wherein said phase calculating step comprises a step of calculating an arcus tangens value of a complex conjugate of said ninth signal.

30. A method according to any one of claims 24 to 29, wherein the step of estimating said frequency offset comprises a step of assigning an fine frequency offset value dependant on the value of said coarse frequency offset according to the following function:

$$\varepsilon = \alpha \qquad ; \text{if } (-0.1)/4 \leq \beta \leq (0.1)/4 \qquad (R1)$$

$$\varepsilon = \alpha \qquad ; \text{if } \alpha \geq 0 \text{ and } (0.1)/4 < \beta < (0.9)/4 \qquad (R2)$$

$$\varepsilon = 1 + \alpha \qquad ; \text{if } \alpha < 0 \text{ and } (0.1)/4 < \beta < (0.9)/4 \qquad (R3)$$

$$\varepsilon = 1 + \alpha \qquad ; \text{if } \beta \geq (0.9)/4 \qquad (R4)$$

$$\varepsilon = -1 + \alpha \qquad ; \text{if } \alpha \geq 0 \text{ and } (-0.9)/4 < \beta < (-0.1)/4 \qquad (R5)$$

$$\varepsilon = \alpha \qquad ; \text{if } \alpha < 0 \text{ and } (-0.9)/4 < \beta < (-0.1)/4 \qquad (R6)$$

$$\varepsilon = -1 + \alpha \qquad ; \text{if } \beta \leq (-0.9)/4 \qquad (R7)$$

EPO-BERLIN

1 6 -07- 2002

Abstract

The IEEE 802.11a standard makes use of the Orthogonal Frequency Division Multiplex (OFDM) transmission scheme. The main feature of the OFDM is that the information stream is not transmitted into a single carrier, but is divided into several sub-carriers, each transmitting at a much lower rate. Furthermore, all these sub-carriers are *orthogonal*, i.e. they overlap their spectra but without causing mutual interference.

The fact that the different sub-carriers overlap their spectra makes specially difficult one of the main operations at the receiver: synchronization. In this standard, the information is not transmitted continuously, but into bursts. Each burst contains a single frame compound of different OFDM symbols. At the beginning of the frame, four preamble symbols are transmitted.

The synchronization process is data-aided, i.e. is based on the digital processing of the preamble symbols, and is responsible to detect the incoming frame as well as to estimate possible frequency errors and to provide a reference channel estimation to the channel estimation block.

We propose a low-power synchronizer structure for the IEEE 802.11a standard, able to estimate frequency offsets in the range ±468 kHz with very simple and effective frame detection and timing synchronization.

Fig.: 17

16 samples = 0.8 μs

| t_1 | t_2 | t_3 | t_4 | t_5 | t_6 | t_7 | t_8 | t_9 | t_{10} | C | D | A | B | C | D | A | B | C | D |

16 μs
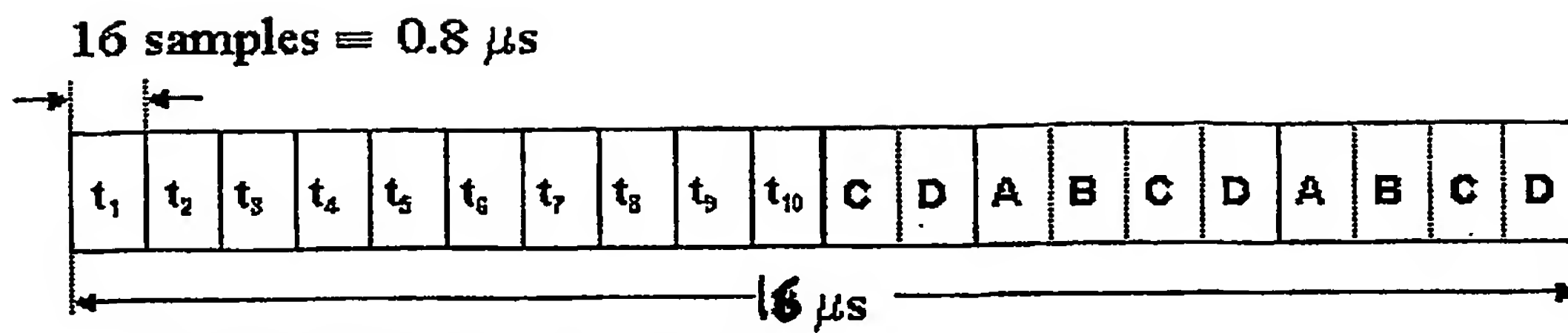
Figure 1. IEEE 802.11a preamble structure.

$y_{OFF}(n)$ → z^{-Nd} / $(.)^*$ → ⊗ → MOVING AVERAGE Navg → $J(k)$ → $|.|^2$ → $|J(k)|^2$
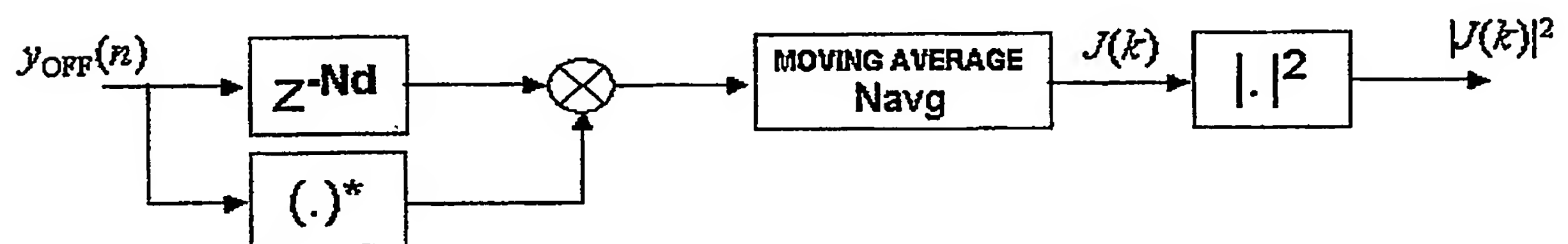
Figure 2. General autocorrelator scheme.

$Re\{in\}$ / $Im\{in\}$ → z^{-1} $(N_{avg}-1)$ → ... → z^{-1} (0) → + / − → $Re\{out\}$ / $Im\{out\}$ — REGISTER
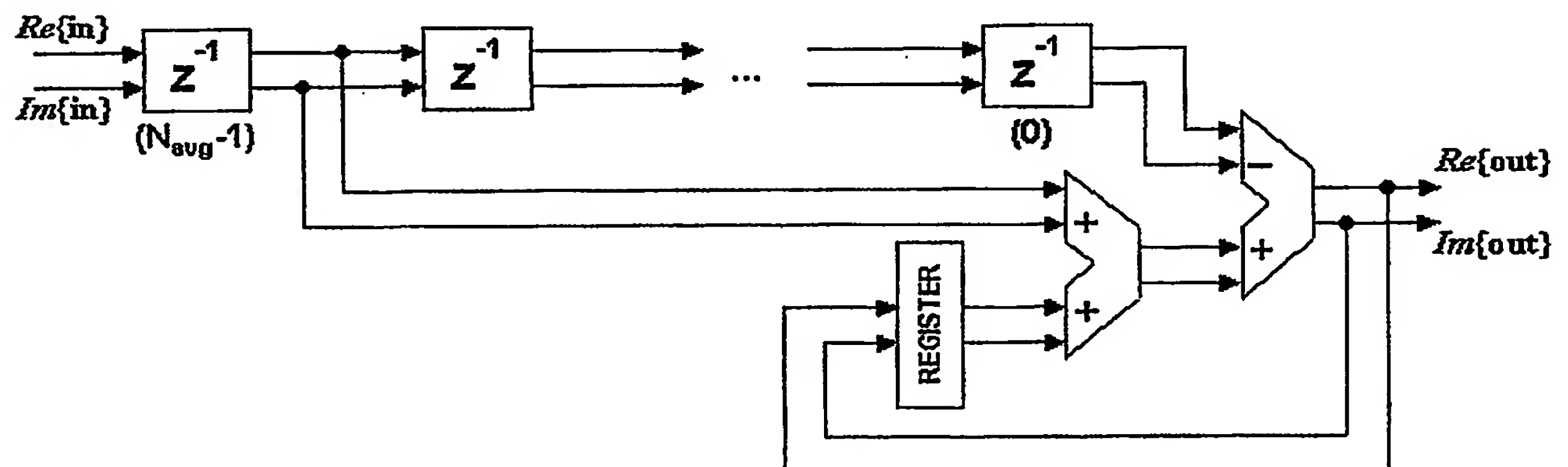
Figure 3. Detailed scheme of the *moving average* block in the autocorrelator.

12
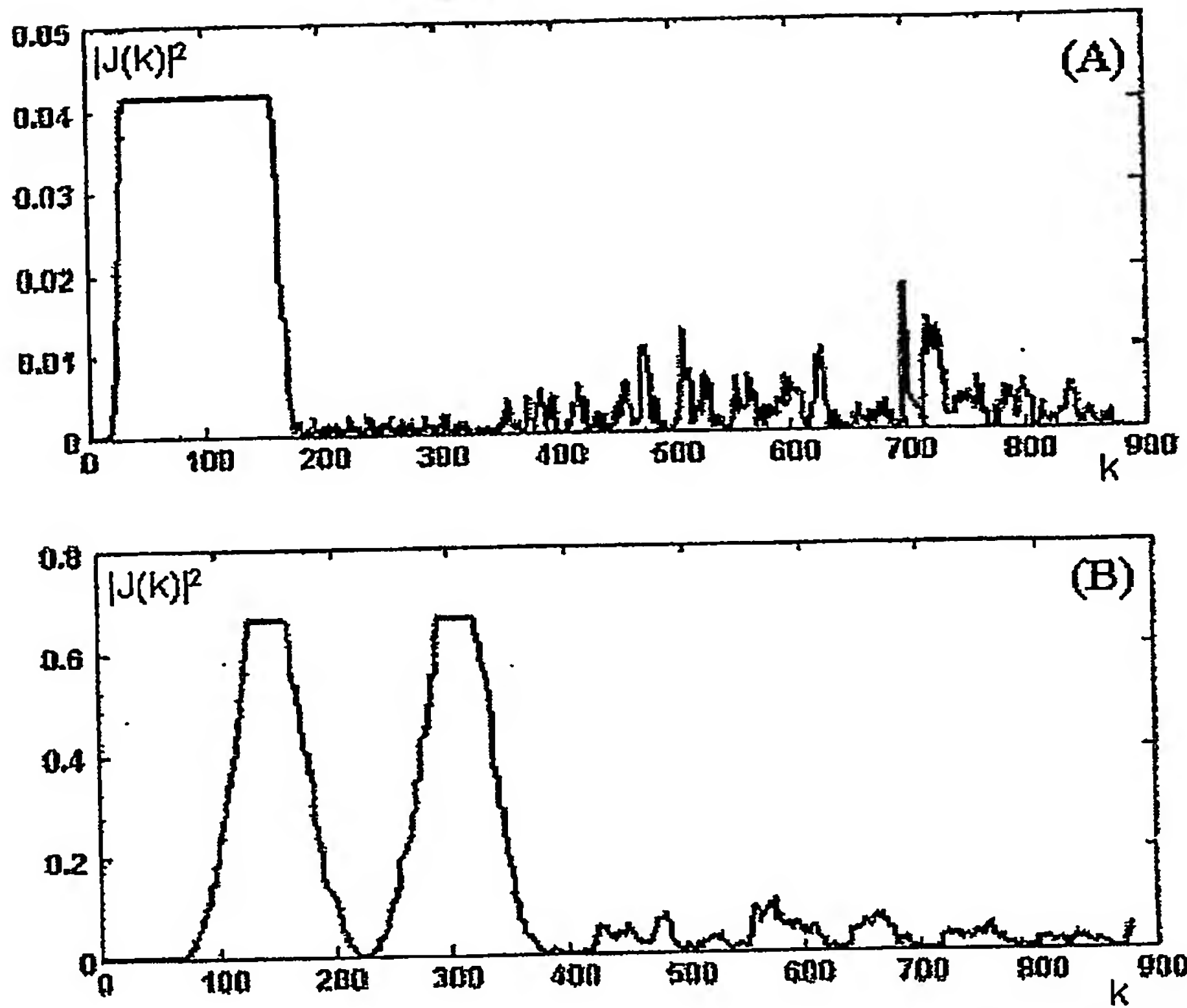
Figure 4. Autocorrelation applied to the IEEE 802.11a preambles: (A) Nd=16, Navg=16; (B) Nd=64, Navg=64.
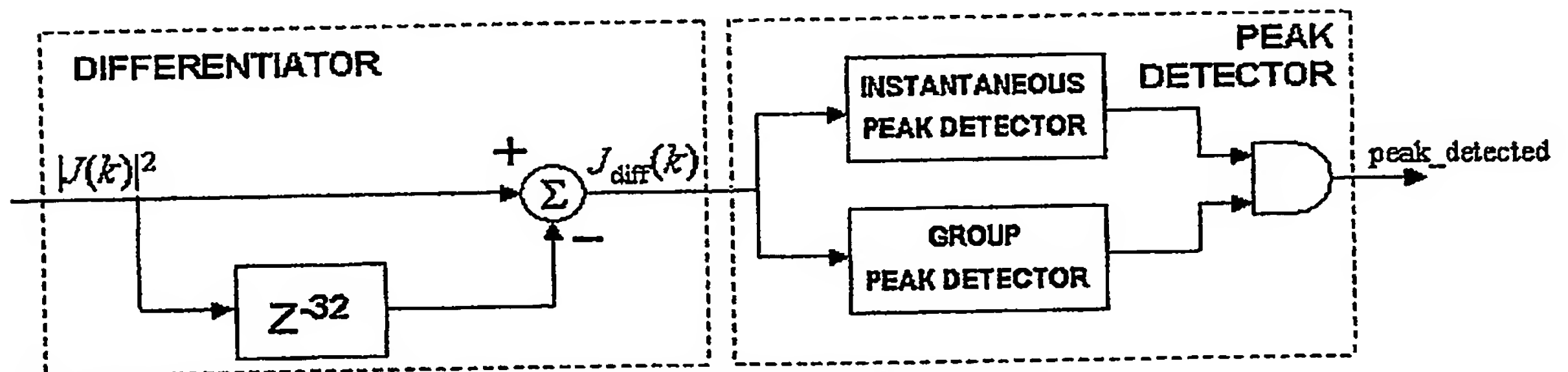
Figure 5. Scheme for the differentiator and peak detector blocks.

13
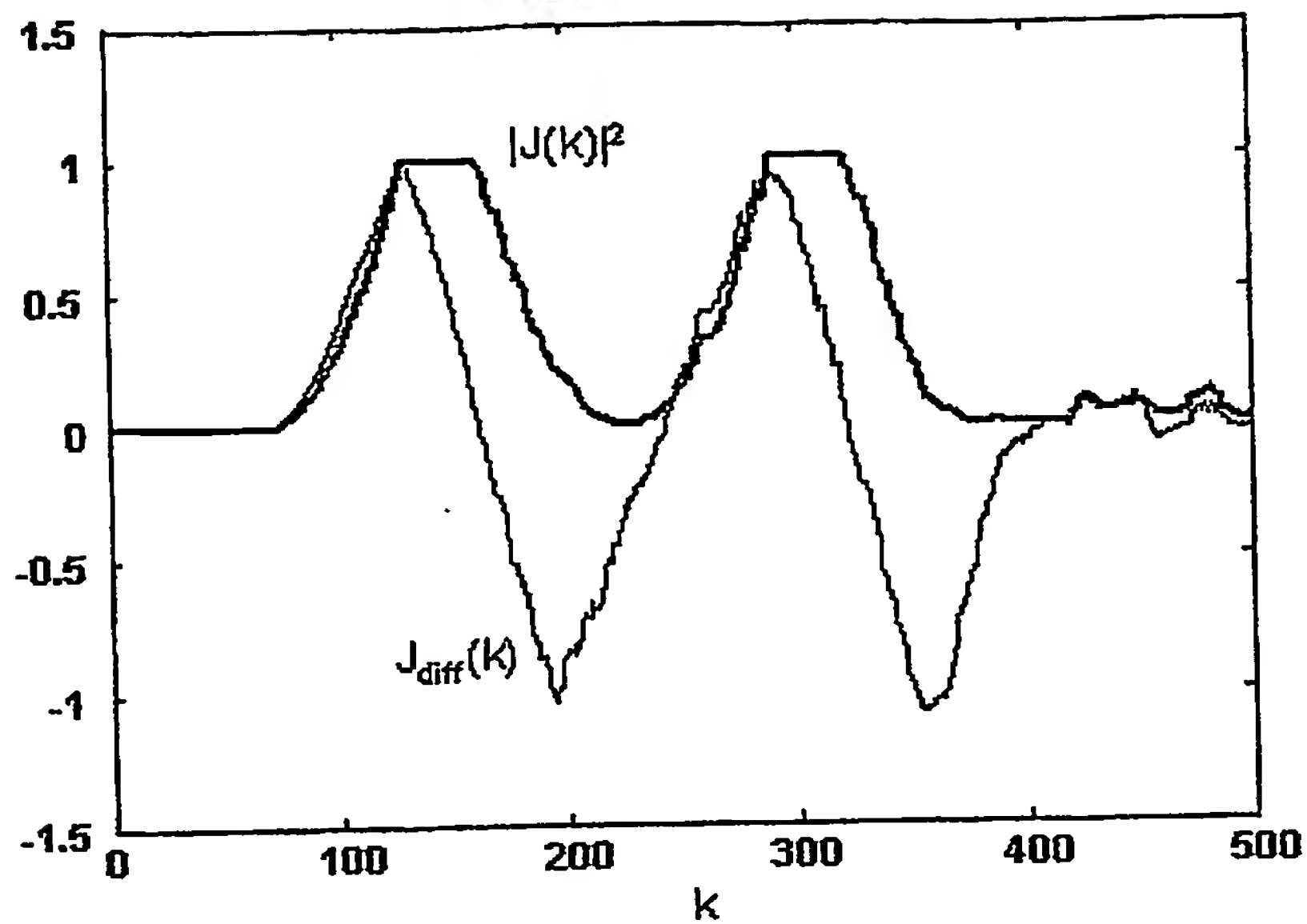
Figure 6. Normalized signal at the output of the differentiator block.
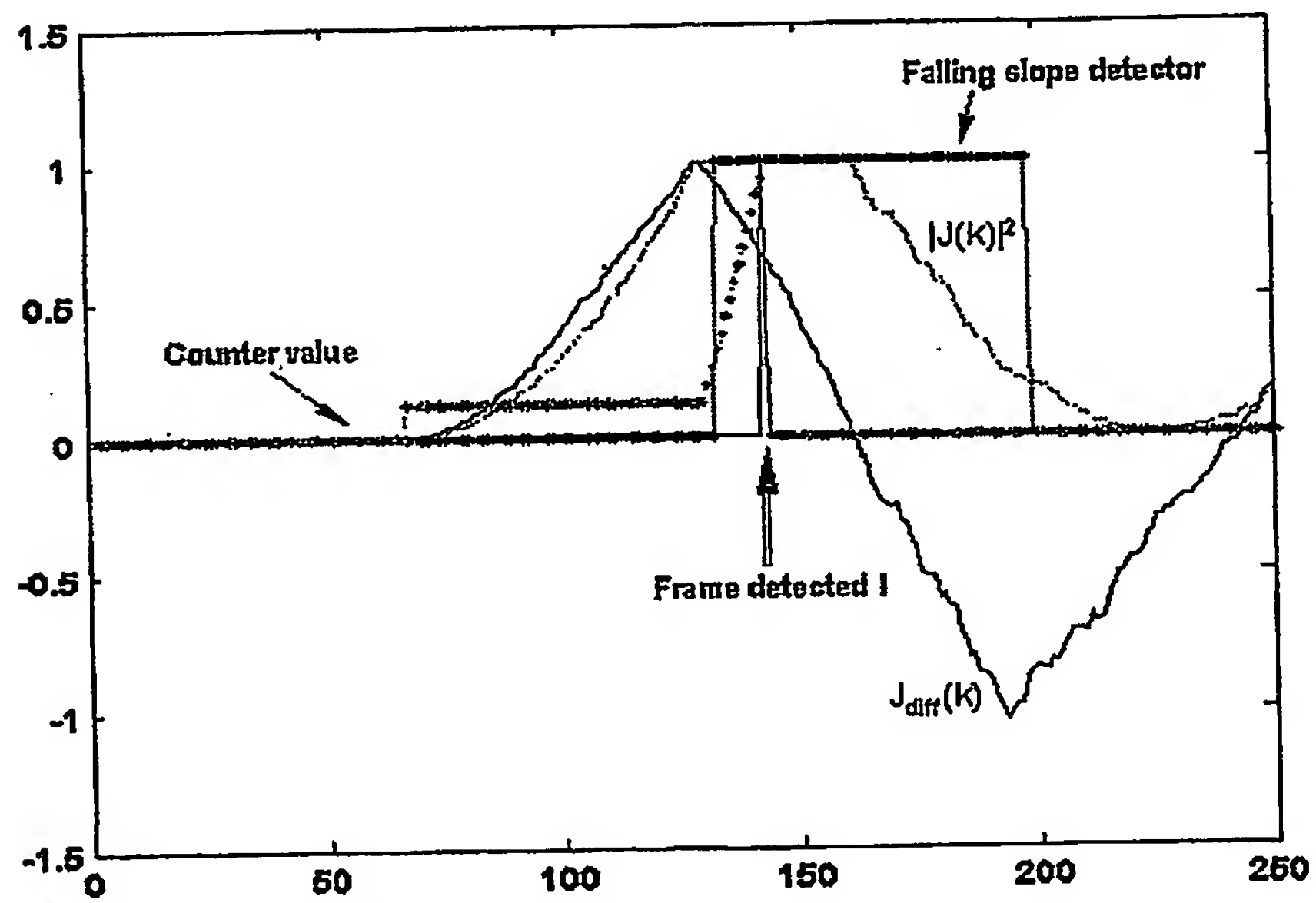
Figure 7. Frame detection procedure.

14
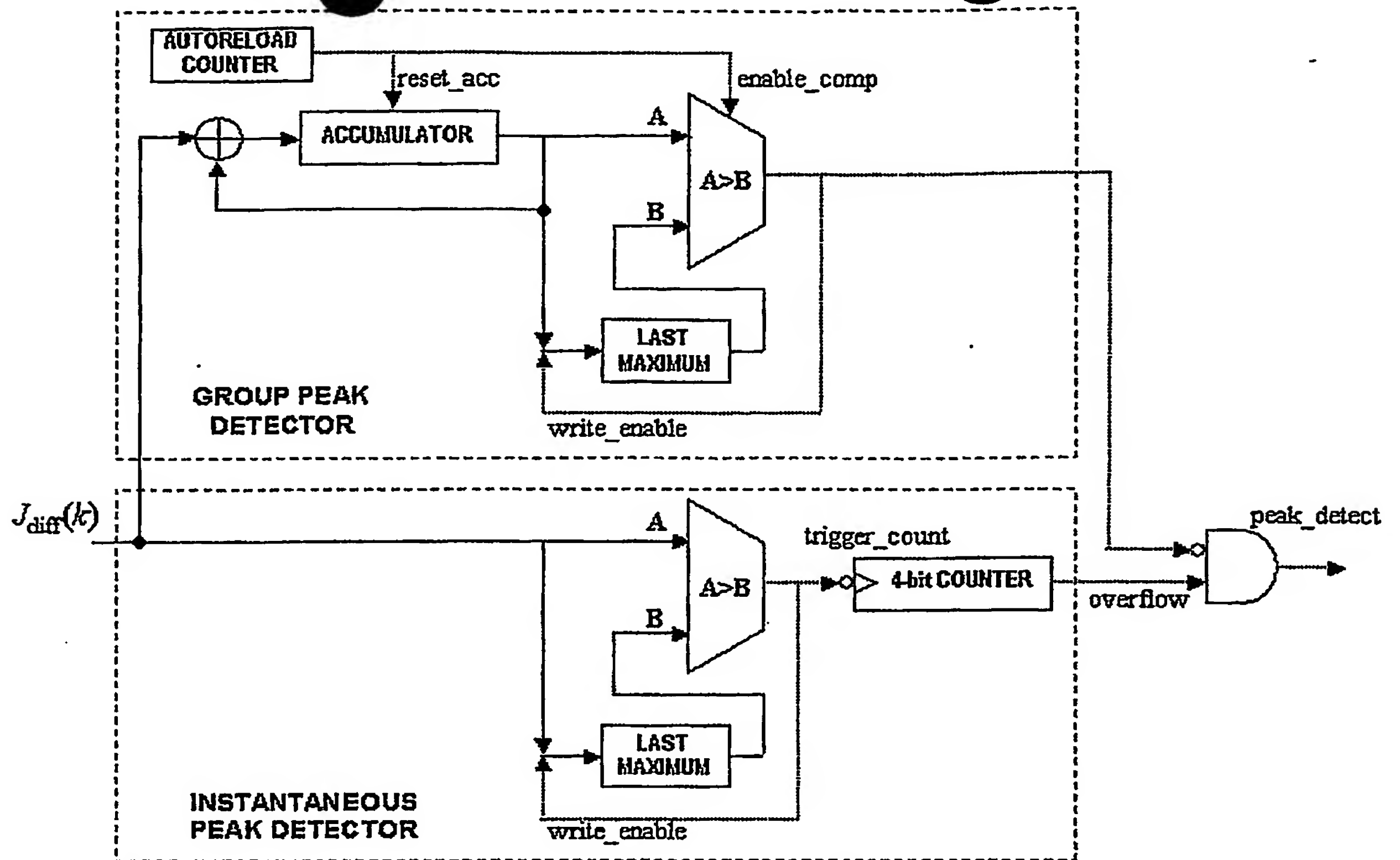
Figure 8. Actual implementation of the peak detector.
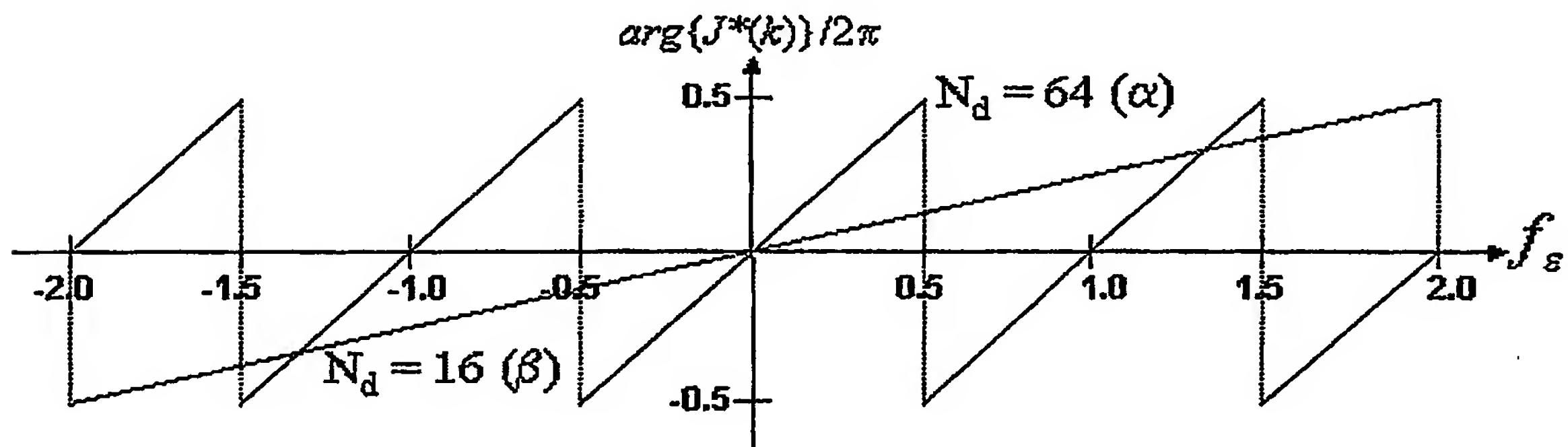
Figure 9. Dependency of the normalized phase of J*(k) with respect to the actual frequency offset (f_ε) and the selected autocorrelation delay N_d.
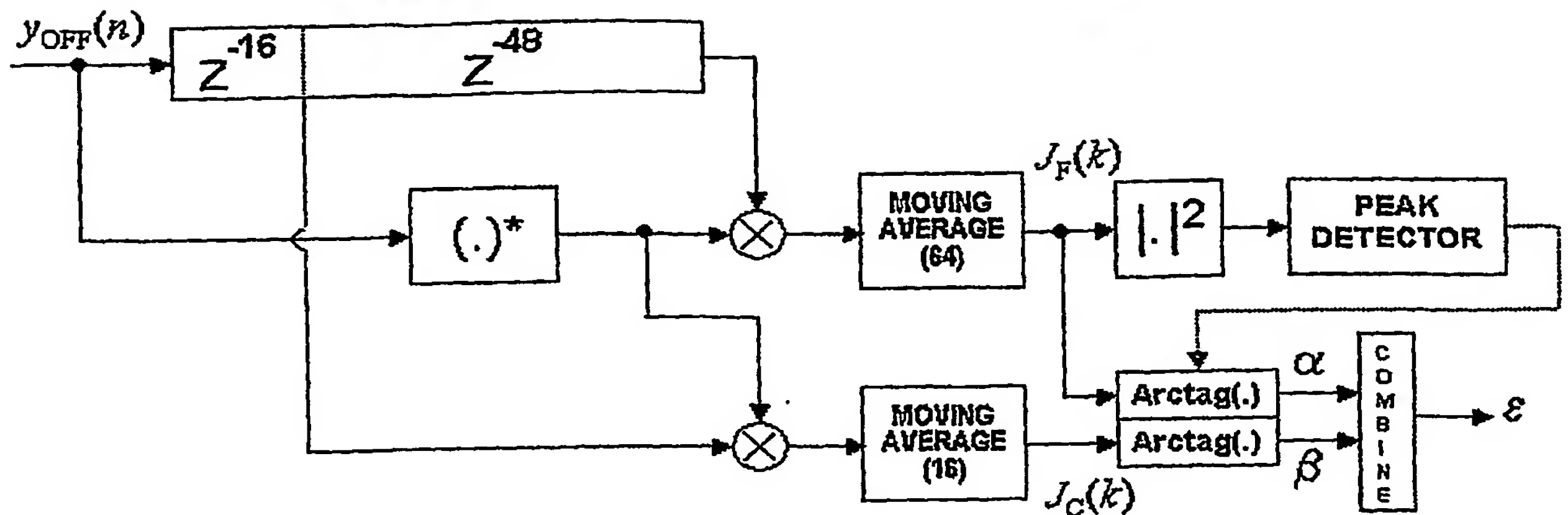
Figure 10. Scheme for the frequency offset estimator.
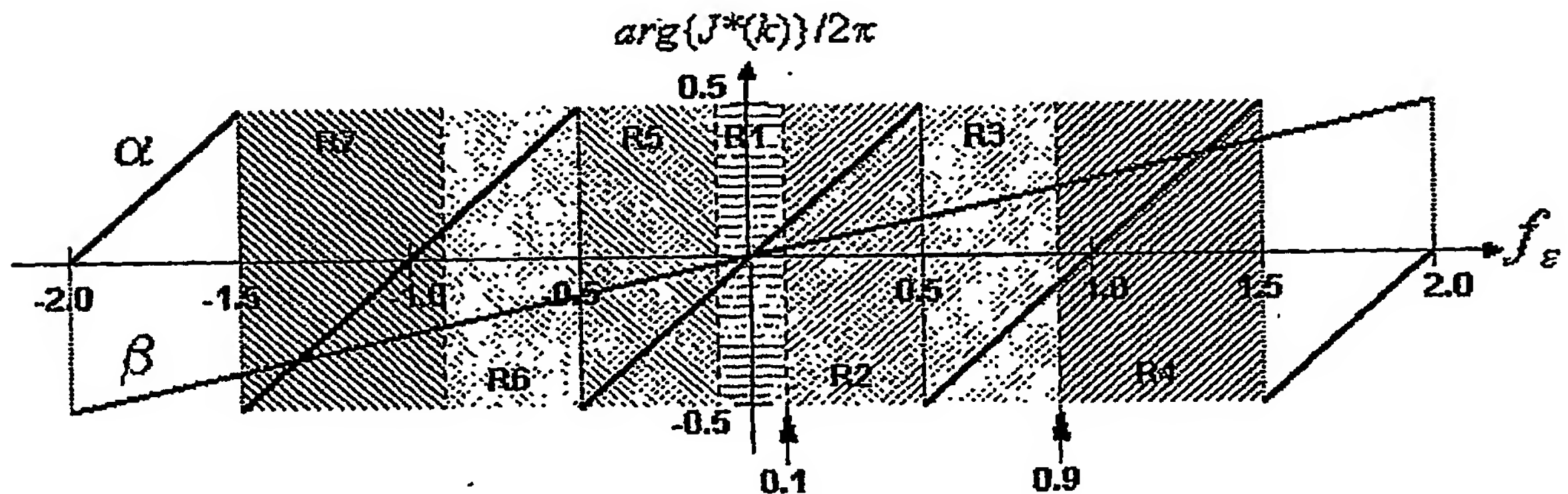
Figure 11. Decision regions for the frequency offset estimation.
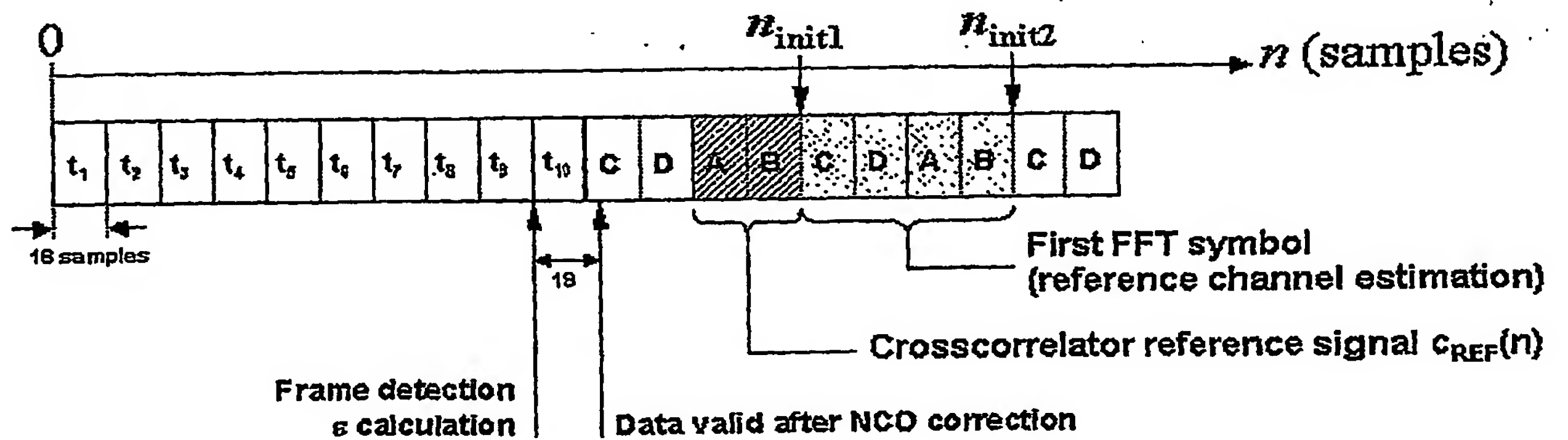
Figure 12. Timing scheme for the different operations realized with the preamble symbols.

16
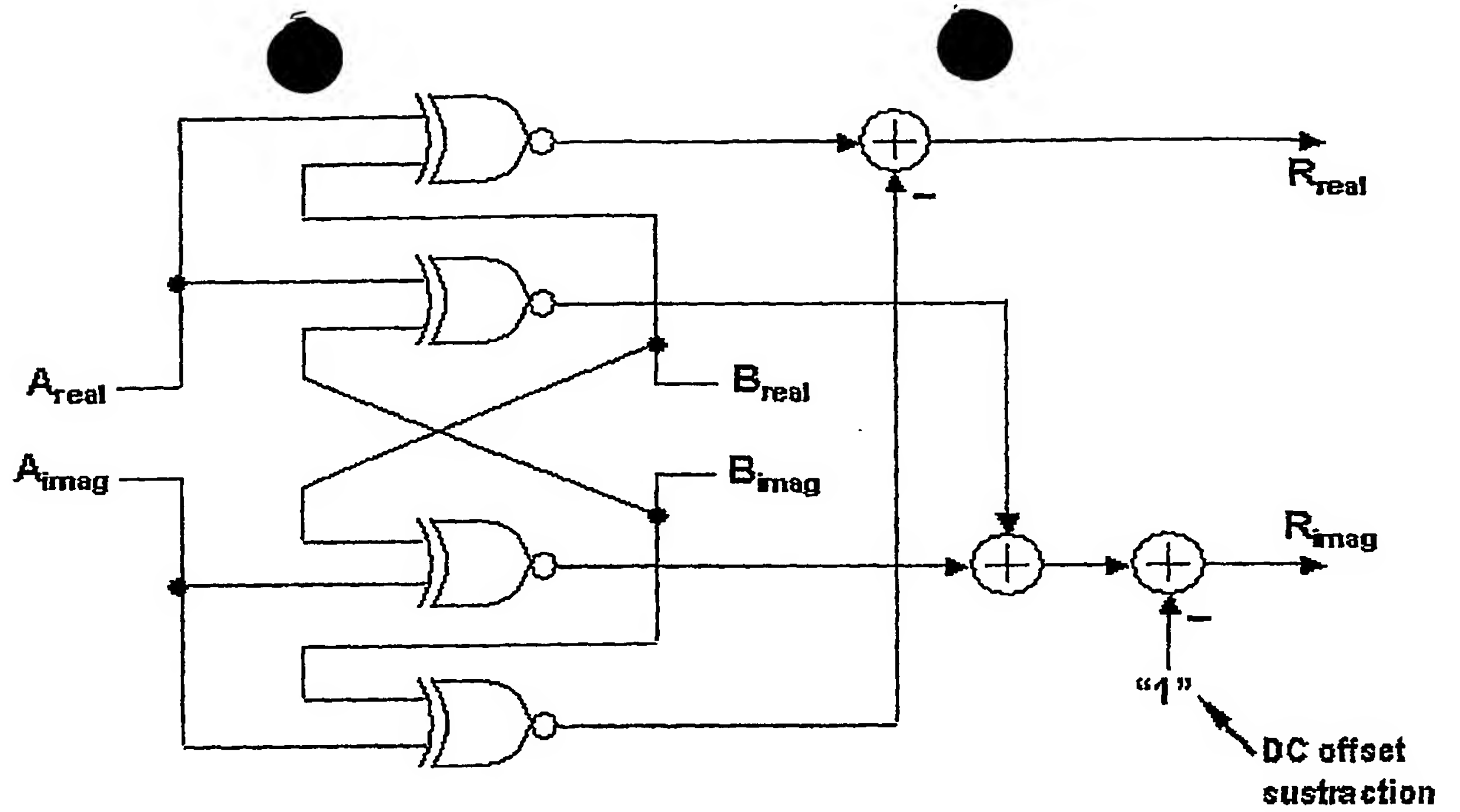
Figure 13. Scheme of the XNOR-based complex multiplier.

Figure 14. Simplified XNOR-based complex multiplier when: (A) $B_{real}=0$, $B_{imag}=0$; (B) $B_{real}=1$, $B_{imag}=1$; (C) $B_{real}=1$, $B_{imag}=0$; (D) $B_{real}=0$, $B_{imag}=1$.
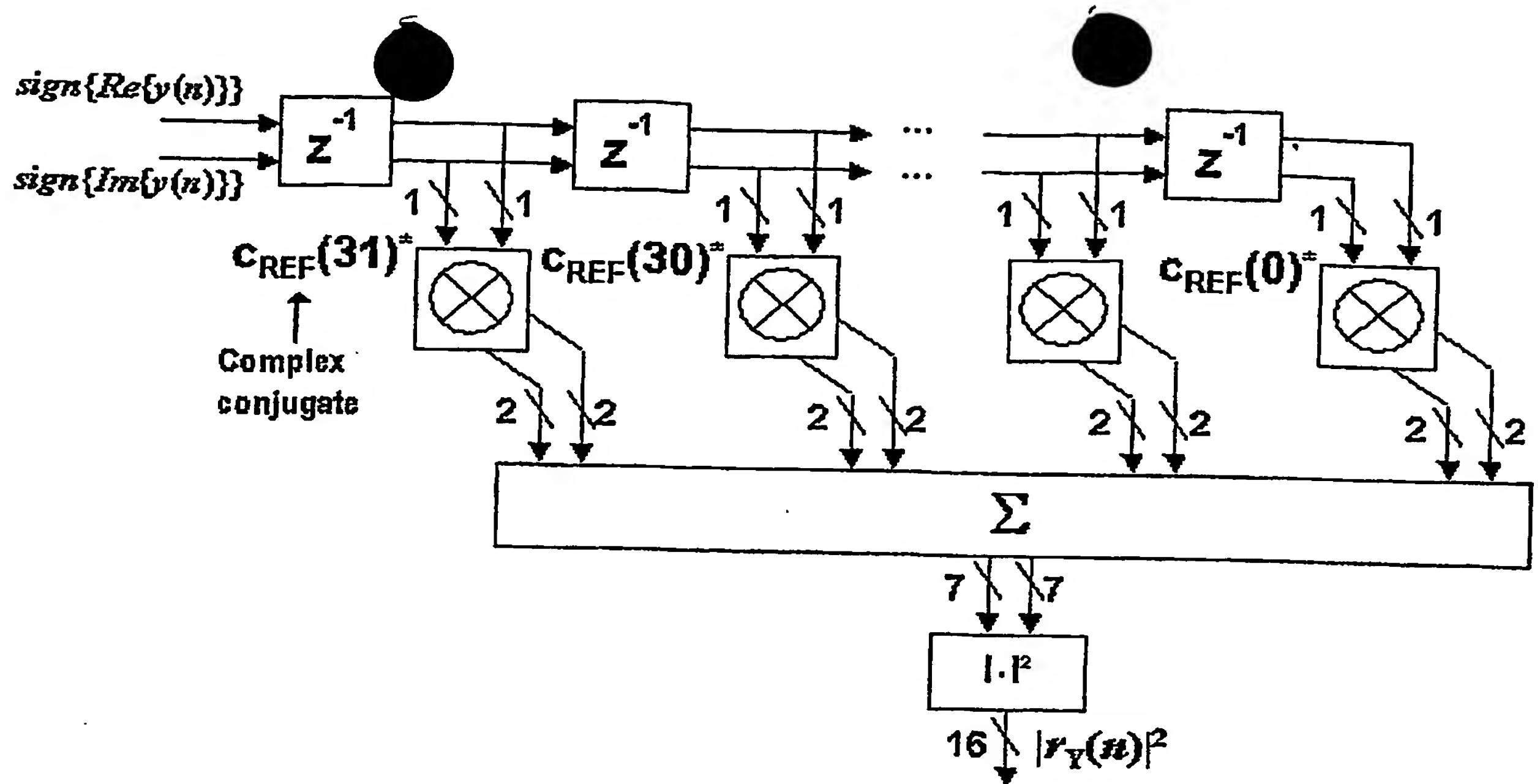
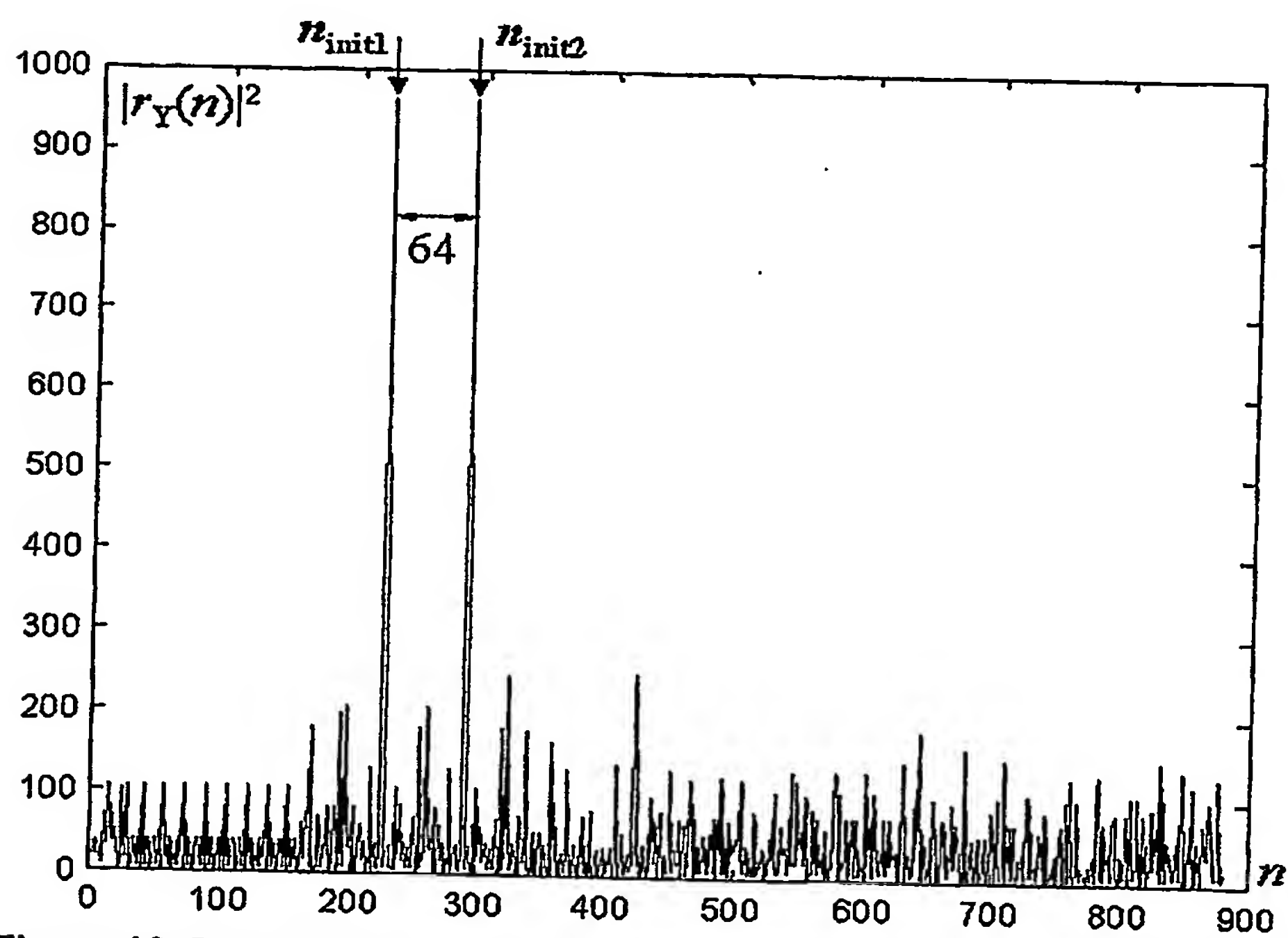Figure 15. Structure of the crosscorrelator considering the simplified XNOR-based architecture.

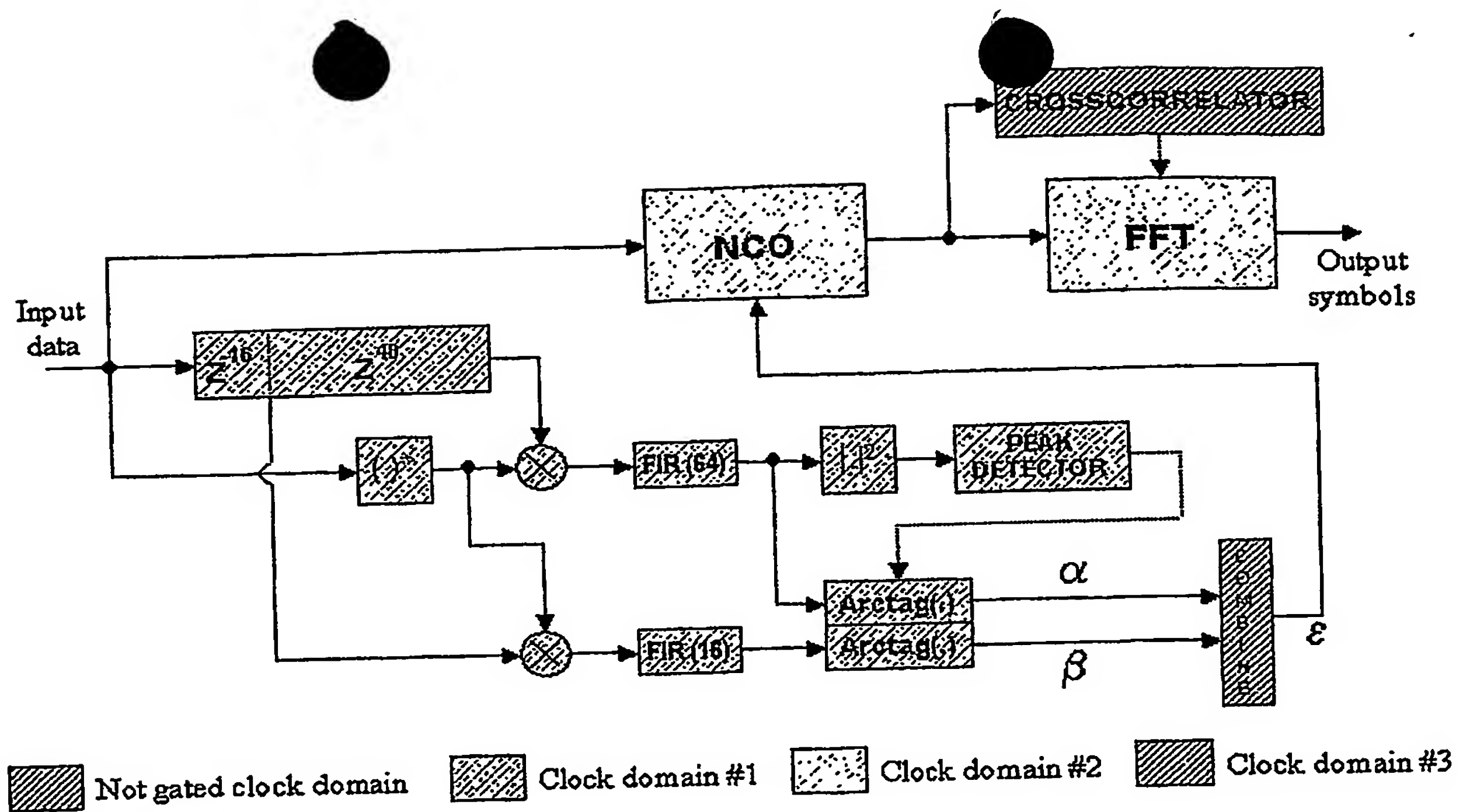Figure 16. Results after crosscorrelation of $c_{REF}(n)$ with the preamble symbols.

Figure 17. General scheme of the synchronizer.

(A)

CORDIC Inputs: $x0, y0, \varphi$
CORDIC Outputs: $x1, y1$

(B)

CORDIC Inputs: $x0, y0$
CORDIC Outputs: $\varphi, \sqrt{(x0)^2+(y0)^2}$

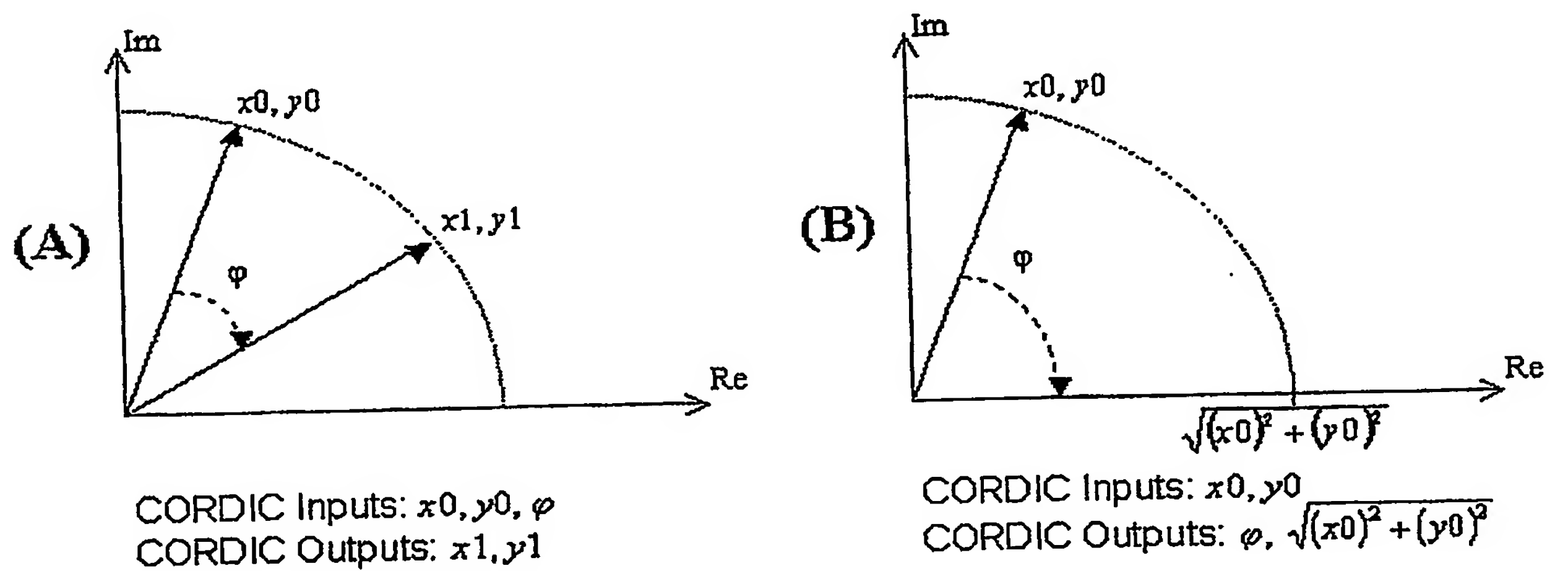**Figure 18.** Working principle of the circular CORDIC algorithm: (A) Rotational mode; (B) Vectoring mode.